



PATENT ABSTRACTS OF JAPAN

(11) Publication number: **11298526 A**

(43) Date of publication of application: 29 . 10 . 99

(51) Int. Cl.

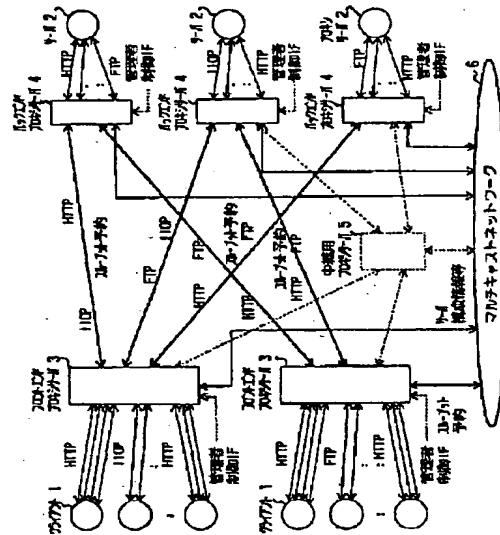
H04L 12/56
G06F 13/00(21) Application number: **10102361**(71) Applicant: **FUJITSU LTD**(22) Date of filing: **14 . 04 . 98**(72) Inventor: **YAMANAKA HIDEKI**(54) **DYNAMIC SERVER THROUGHPUT INVERSE
DIVISION RESERVATION SYSTEM**

COPYRIGHT: (C)1999,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To optimize a server load by introducing a resource reservation function to a system where clients and servers are interconnected via a network.

SOLUTION: Each front end proxy server 3 that converts a web protocol into a protocol with a resource reservation function for transaction processing between clients 1 and servers 2 is placed in the vicinity of the clients and back end proxy servers 4 are placed in front of the servers 2. Each front end proxy server 3 connects to plural back end proxy servers 4 to increase total throughput of the back end proxy servers 4 by distributing the load. Each of the back end proxy servers 4 measures the throughput of the server 2 and reserves the throughput dynamically to each client 1 depending on the state of load based on the server load and the characteristic of the server throughput measured in advance.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-298526

(43) 公開日 平成11年(1999)10月29日

(51) Int.Cl.⁸

識別記号

F I

H 0 4 L 12/56

H 0 4 L 11/20

1 0 2 C

G 0 6 F 13/00

3 5 1

G 0 6 F 13/00

3 5 1 A

審査請求 未請求 請求項の数 4 O L (全 11 頁)

(21) 出願番号

特願平10-102361

(22) 出願日

平成10年(1998)4月14日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 山中 英樹

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 弁理士 長澤 俊一郎 (外1名)

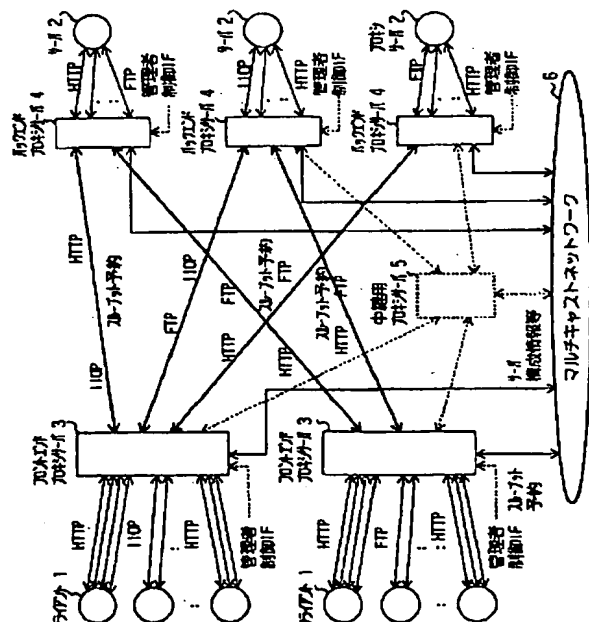
(54) 【発明の名称】 動的サーバスループット逆分割予約システム

(57) 【要約】

【課題】 クライアントとサーバがネットワークを介して接続されているシステムにリソース予約機能を導入し、サーバ負荷の最適化を図ること。

【解決手段】 クライアント1とサーバ2との間でWebのプロトコルをトランザクション処理向けのリソース予約機能を持つプロトコルに変換するフロントエンド・プロキシサーバ3をクライアントの近辺に置き、サーバ2の前にバックエンドのプロキシサーバ4を置く。各フロントエンド・プロキシサーバ3からこの複数のバックエンド・プロキシサーバ4に接続を張り、負荷を分散させることでトータルのバックエンド・プロキシサーバ4のスループットを増大する。上記バックエンド・プロキシサーバ4で、サーバ2のスループットを計測し、予め測定したサーバ負荷と、サーバスループットの特性に基づき、負荷状況に応じて動的に各クライアント1に対してスループットを予約する。

本発明の原簿構成図



【特許請求の範囲】

【請求項1】 クライアントとサーバがネットワークを介して接続されているネットワーク・システムにおけるサーバスループット予約システムであって、クライアントとサーバ間のクライアントの近傍にリソース予約機能を持つフロントエンド・プロキシサーバを設けるとともに、サーバの近傍にリソース割り当てを行うバックエンド・プロキシサーバを設け、上記バックエンド・プロキシサーバで、サーバのスループットを計測し、予め測定したサーバ負荷と、サーバスループットの特性に基づき、負荷状況に応じて動的に各クライアントに対してスループットを予約することを特徴とする動的サーバスループット逆分割予約システム。

【請求項2】 クライアントとサーバがネットワークを介して接続されているネットワーク・システムにおけるサーバスループット予約システムであって、クライアントとサーバ間のクライアントの近傍にリソース予約機能を持つフロントエンド・プロキシサーバを設けるとともに、サーバの近傍にリソース割り当てを行うバックエンド・プロキシサーバを設け、上記バックエンド・プロキシサーバで、サーバのスループットを計測し、指定されたサーバ負荷とサーバスループット・モデルに基づき、負荷状況に応じて動的に各クライアントに対してスループットを予約することを特徴とする動的サーバスループット逆分割予約システム。

【請求項3】 WebクライアントとWebサーバがネットワークを介して接続されているネットワーク・システムにおけるサーバスループット予約システムであって、WebクライアントとWebサーバ間のWebクライアントの近傍にリソース予約機能を持つフロントエンド・プロキシサーバを設けるとともに、Webサーバの近傍にリソース割り当てを行うバックエンド・プロキシサーバを設け、上記バックエンド・プロキシサーバで、Webサーバのスループットを計測し、Webクライアントからのスループット予約に基づき、負荷状況に応じて動的に各Webクライアントに対してスループットを予約することを特徴とする動的サーバスループット逆分割予約システム。

【請求項4】 WebクライアントとWebサーバがネットワークを介して接続されているネットワーク・システムにおけるサーバスループット予約システムであって、WebクライアントとWebサーバ間のWebクライアントの近傍にリソース予約機能を持つフロントエンド・プロキシサーバを設けるとともに、Webサーバの近傍にリソース割り当てを行うバックエンド・プロキシサーバを設け、上記バックエンド・プロキシサーバで、Webサーバの

スループットを計測し、予め測定したサーバ負荷と、サーバスループットの特性に基づき、負荷状況に応じて動的に各クライアントに対してスループットを予約することを特徴とする動的サーバスループット逆分割予約システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、インターネット、イントラネット等で使われるWeb通信の仕組みにリソース予約機能を効果的に付加することができる動的サーバスループット逆分割予約システムに関する。

【0002】

【従来の技術】現在のWeb通信は、ハイパテキスト内に記述されたオブジェクトに対し、1本ずつのTCPコネクションを張り、これを使って一つずつダウンロードしながら画面に表示する。しかし、逐次にダウンロードを実行すると、1本ずつのコネクションが帯域幅を全て使い尽くす程高速ではないので、効率が悪い。また、快適性のために、複数のオブジェクトを同時にダウンロードすることで、画面を同時描画することが望ましい。これらの解消手段として、ハイパテキスト内に記述された複数オブジェクトに対して並列にTCPコネクションを張り、同時にダウンロードを実行することが行われる。

【0003】この方法は、各クライアントにとってはユーザを快適にするための戦略であるが、逆にサーバ側から見れば単なるリソースの無駄使いである。しかも、悪いことに並列に張るコネクションを多くする程ユーザは他のユーザより快適になるので、クライアントが競い合ってサーバのリソースを浪費することになる。サーバは、過負荷状態に入ってしまうと極端に性能が低下するので、快適性のために浪費したリソースのためにかえってユーザは不快な状態に陥ることになってしまう。サーバ性能を最大限に引き出すためには、リソースの消費を一定の範囲に抑え続けることが要件であり、その戦略のためにはユーザに対するリソースの割り当てを行うのが通常である。このようにユーザの快適性のための戦略とサーバ性能を最大に保つための戦略が、相互に矛盾していて、現実にはインターネットはWebの通信で慢性的な輻輳状態を引き起こしている。

【0004】

【発明が解決しようとする課題】しかしながら、現在のWebの仕組みは、サーバが不特定多数のユーザを扱わなければならないので、予めユーザ毎にリソースを割り当てることができない。また、Webのインフラであるインターネットは、学術ネットワークを基盤にしたこととインフラの費用を安く抑えるために、リソース割当の機能が省略されている。最近、インフラであるインターネットに帯域リソース予約機能を導入する動きもあるが、Webの仕組みはサーバ上のリソースの予約も同時に行う必要があるので、不十分である。

【0005】Web環境でリソース予約機能を導入するとしても、以下の問題がある。すなわち、通常のWebアクセスは、サーバとクライアントが直接TCPコネクションを確立し、1度に数KB程度のオブジェクトをHTTPプロトコル等のWebのプロトコルでアップロード、またはダウンロードし、最後にコネクションを消去すること（トランザクション処理）で行われる。クライアントのリソース予約は、Webのプロトコルではできないので、Webプロトコル以外の別プロトコルを併用する必要がある。また、平均的にWebのTCPコネクションは短命なので、各TCPコネクション単位でリソース予約を行っても、予約の効果が現れる前にコネクションが消去されることが多いので、単にオーバーヘッドが大きくなるだけである。本発明は上記事情に鑑みなされたものであって、その目的とするところは、クライアントとサーバがネットワークを介して接続されているシステムにリソース予約機能を導入し、サーバ負荷の最適化を図ることである。

【0006】

【課題を解決するための手段】図1は本発明の原理構成図である。同図においては、一般化のために、特にWebサーバ、クライアントと明記していないが、本発明は、基本的にWeb環境でのリソース予約システムとして最適に機能する。クライアント1として、通常のWebブラウザだけでなく、Webプロキシサーバも想定している。フロントエンドプロキシサーバ3とバックエンドプロキシサーバ4の間に中継用予約サーバ5を置くこともできる。

【0007】Webのプロトコルではリソース予約が出来ず、リソース予約機能を導入するには、次のことが要求される。

Webのプロトコルとは別のプロトコルのコネクションで通信する必要がある。

通常、そのために直接ユーザのクライアントプログラムを改変することは、コストの面、セキュリティの面で許されない。

HTTPプロトコルは元々トランザクション処理に向かず性能が悪い。

以上のことから、本発明においては、図1に示すように、クライアントプログラム1とサーバ2との間でWebのプロトコルをトランザクション処理向けのリソース予約機能を持つプロトコルに変換するフロントエンドのプロキシサーバ3をクライアントの近辺に置く。また、このプロトコルのためにサーバ2を改変するのは既存のシステムとの互換性が悪く、開発効率も悪いので、サーバ2の前にもバックエンドのプロキシサーバ4を置く。

【0008】このバックエンドのプロキシサーバ1台をサーバ1台に対応付けることで、サーバ2上で直接リソース割り当てを行うのと同等の処理をバックエンドのプロキシサーバ4で行うことができる。バックエンド・プ

ロキシサーバ4のスループットがサーバスループットより小さいときには、バックエンド・プロキシサーバ複数台をサーバ2に対応付け、各プロキシサーバの性能を考慮してサーバスループットを分割して割り当てる。そして、各フロントエンド・プロキシサーバ3からこの複数のバックエンド・プロキシサーバ4にコネクションを張り、負荷を分散させることでトータルのバックエンド・プロキシサーバ4のスループットを増大することができる。さらに、規模の大きなシステムではフロントエンド・プロキシサーバ3とバックエンド・プロキシサーバ4の間に中継用キャッシュ機能付きプロキシサーバ5を置くこともできる。

【0009】ここで導入するプロトコルは、Webのプロトコルの複数コネクションのリソース予約を同時に行うためと、トランザクション向きにHTTPプロトコルを最適化するため、Webのプロトコルをカプセル化しリソース予約が可能な多重化プロトコルである。プロキシサーバ間をTCPコネクションで結び、この間をこの多重化プロトコルで通信することで、TCPコネクション中に同時に複数の仮想的なWebコネクションを張るとともに、リソース予約の通信も同時に行うことができる。また、マルチキャストネットワーク6を設けることにより、各フロントエンド・プロキシサーバ3、バックエンド・プロキシサーバ4、中継用プロキシサーバ5は、上記マルチキャストネットワーク6を介して他サーバの構成情報等を得ることができる。

【0010】スループット予約は以下のように行うことができる。

(1) 予め測定したサーバの負荷とサーバスループット（サーバ側のトータルスループット）の特性を使って、サーバが過負荷状況に陥らないように、サーバが各クライアントのスループットを制約するように予約することで、クライアントからの要求によるリソースの浪費を抑え込む。また、クライアントスループット予約時にユーザを同定することで、各ユーザへのリソース割当を公平にし、ユーザ間のリソースの奪い合いを無効にする。逆に、特権を持つクライアントにはサーバスループットの予約を許し、他のクライアントによるリソースの奪取を防ぐ（請求項1）。

【0011】例えば、一般に、Webサーバの負荷（同時コネクション数）とサーバスループットの特性は、図2に示すように、ある一定の負荷が掛かるまで、サーバスループットがほぼ線形に増える。この点（S点）を越えるとサーバスループットは増加せず、ほとんど一定の値（最大スループット）を保つが、さらに負荷が大きくなると（E点）、逆にサーバスループットの低下が始まる。つまり、S点に達する前の負荷が軽い状態では、クライアントのリソースを予約する必要がない。S点からE点の間は、負荷の変動に付してサーバスループットが一定であるので、各ユーザがサーバスループットを奪い

5

合う競合状態になり、クライアントのリソース予約が必要である。E点を越える負荷状況は、サーバの危機的な状態なので、クライアントへの新たなリソース割当を停止して、負荷が軽くなるのを待たなければならない。

【0012】リソース予約の仕組みは、サーバの直前のバックエンド・プロキシサーバ4で、サーバ2のスループットを計測し、サーバの負荷とサーバスループットのモデルに従いクライアント側のフロントエンド・プロキシサーバ3、中継プロキシサーバ5にスループットを以下のようにして予約する。

S点までの負荷状況では、クライアント側の全てのスループット予約を解除する。

S点からE点では各フロントエンド・プロキシサーバ3と中継プロキシサーバ5に予め与えられた重み、あるいは特権を持つフロントエンド・プロキシサーバと中継プロキシサーバが予約した重みに従ったサーバスループットの分割予約を行う。バックエンドプロキシサーバ4は、各フロントエンド・プロキシサーバ3、中継プロキシサーバ5にスループット予約を行うとともに、それらとの多重化コネクションのI/O頻度を重みに従った比で行う。

サーバ負荷がE点を越えた場合は、クライアント側のフロントエンド・プロキシサーバ3に新たな仮想Webコネクションを張らないようにリソースを予約する。サーバ負荷がE点より下がれば、新たな仮想Webコネクションを張れるようにリソース予約を解除する。

中継プロキシサーバ5は、予約されたスループットをそのフロントエンド・プロキシサーバに対して、同様に分割予約して行く。

【0013】各フロントエンド・プロキシサーバ3は、そこに接続したコネクションのユーザを同定し、各ユーザのトータルのスループットが同じになるように各コネクションにスループットを割り当て、その値を越えないようにI/Oをスケジューリングする。以上のようにサーバ負荷とサーバスループットは、原点からS点への線形領域の線分、S点からE点への水平な飽和領域の線分でモデル化できるので、これに基づき各クライアントのスループットを制限すれば、サーバスループットを最適に保持できる。

【0014】(2)サーバ負荷とサーバスループット特性は、実際のサーバのものである必要はない。また、サーバの特性とは違ったものを指定することができ、予め指定されたサーバ負荷とサーバスループット・モデルに基づき、負荷状況に応じて動的に各クライアントに対してスループットを予約することもできる(請求項2)。サーバ負荷とサーバスループットの特性を測定する作業は、測定システム、装置の設置、設定、実際の測定等複雑であるが、上記のようにすることにより、他の既に測定された同程度の性能と考えられるサーバ特性で代用させることが可能となる。また、実際にこの予約システム

6

を運用したときに得られたシステムの統計データから、S点、E点をチューニングすることもできる。

【0015】(3)図1に示すように、予約サーバに、特権を持ったクライアントからの予約サーバへのスループット予約を可能とする管理者制御インタフェースを設けることもできる。このインタフェースは、予め登録されたクライアントからのアクセスだけを許可する機能を持ち、このクライアントに関するスループット予約命令だけを実行できる。予約サーバの管理者は、この管理者制御インタフェースを使って、このサーバの任意のクライアントのスループット予約が可能である。また、予約サーバが備えるデフォルト予約テーブルを書き換えることで設定することができる(請求項3)。Webの仕組みには、サーバのリソースを予約する機能がないが、上記のように予約サーバの管理者制御インターフェースを使えば、間接的にサーバのリソースが予約可能になる。このことにより、全てのWebクライアントを平等に扱うのではなく、特定のWebクライアント、ユーザのグループに優先的にリソースを割り当てることも可能となる。

【0016】(4)上記(1)をWeb通信に適用し、サーバへの負荷をバックエンド予約サーバで常に計測し、その負荷の変動に従ってフロントエンド予約サーバにサーバ負荷を増減させるスループット予約命令を発行する(請求項4)。Webの仕組みには、サーバが自身に掛かる負荷を制御する機能がないが、本発明をWeb通信に適用し、予約サーバがサーバ負荷を常に監視し、クライアント側にサーバ負荷を動的に効果的に増減させる命令を送る仕組みを導入することにより、サーバ負荷を自律的に間接的に制御できる。さらに、サーバの負荷とスループットのモデルを組み込むことにより、Web通信において、サーバのスループットを最適に保持できる。

【0017】

【発明の実施の形態】以下、本発明をWeb通信に適用した場合の実施例を説明する。図3は本発明の実施例のシステムの構成を示す図、また、図4は図3のシステムに中継サーバ50を設けた場合のシステムの構成を示す図である。図3、図4において、30はフロントエンド予約サーバ、40はバックエンド予約サーバであり、予約サーバ30、40は各Webサーバ、Webプロキシサーバ20のフロントエンド、バックエンドとして動作する。フロントエンド予約サーバ30とバックエンド予約サーバ40間は、前記のようにTCPコネクションで結ばれ、この間を多重化プロトコルで通信することで、TCPコネクション中に同時に複数の収束的なWebコネクションを張るとともに、リソース予約の通信も同時に行う。

【0018】フロントエンド予約サーバ30、バックエンド予約サーバ40はマルチキャストネットワーク6に

接続されており、これらのサーバが起動されると次のように動作する。

マルチキャスト通信により他の既に動作しているサーバの情報を獲得し、必要に応じて動的にこれらのサーバとの間に多重化プロトコルで通信可能な接続を張る（ある一定時間リクエストの通信がなければ、接続を消去する）。

Webクライアント10は、リクエストを最も近くにあるフロントエンド予約サーバ30に転送する。

リクエストを受けたフロントエンド予約サーバ30は、Webサーバ20の近くにあるバックエンド予約サーバ40へ多重化プロトコルを使ってリクエストを中継する（もしくは中継サーバ50を介してバックエンド予約サーバ40へリクエストを中継する）。

バックエンド予約サーバ40は、Webサーバ20へリクエストを転送して、返事を受け取る。

バックエンド予約サーバ40、フロントエンド予約サーバ30（もしくは中継サーバ50）は、リクエストが送られて来たパスを逆に辿って、返事をWebクライアント10へ返す。

【0019】ここで、バックエンド予約サーバ40は、Webサーバ20のスループットを観測しながら、前記したようにWebサーバ20の負荷とサーバスループットモデルに従って各フロントエンド予約サーバ30（中継サーバ50）にそのサーバに対するスループットを予約する。各フロントエンド予約サーバ30は、同一のWebサーバに接続している全ての接続のユーザを同定し（同一IPアドレスを持つ接続を1ユーザとする。あるいは、ident等のツールでユーザを同定する）、ユーザ単位で平等になるように各接続にスループットを割り当て、その値を越えないようにI/Oをスケジューリングする。また、中継サーバ50は前記したように、予約されたスループットをそのフロントエンド予約サーバに対して分割予約して行く。

【0020】図5、図6はフロントエンド予約サーバ、バックエンド予約サーバの構成を示す図であり、図5、図6は送受信器A、B、スループット制御装置A、B、多重化装置の左右の位置が入れ代わった点を除き同一である。また、図7は中継サーバの構成を示す図であり、図7は図5、図6から送受信器A、スループット制御装置Aを除去したものである。図5、図6、図7において、31、41は送受信器A、36、46、56は送受信器B、32、42はスループット制御装置Aであり、スループット制御装置Aは、送受信器Aで受信したパケットもしくは送受信器Aから送信するパケットを一時的に保持し、スループット予約情報に基づき順次転送する。また、35、45、55はスループット制御装置Bであり、スループット制御装置Bは、上記と同様、送受信器Bで受信したパケットもしくは送受信器Bから送信するパケットを一時的に保持し、スループット予約情報

に基づき順次転送する。

【0021】34、44、54は多重化装置、33、43、53はパケットスイッチであり、多重化装置34、44、54はパケットスイッチ33、43、53の出力をマルチプレックスしてスループット制御装置Bに送出し、また、スループット制御装置Bの出力をディマルチプレックスしてパケットスイッチ33、43、53に送り出す。38、48、58はスループット予約装置であり、スループット予約装置38、48、58には、スループット予約テーブルが設けられており、スループット予約テーブルには後述するように、システム構成情報（他サーバの負荷状態、サービスの内容等）、予約情報等が格納される。39、49、59はマルチキャスト送受信器であり、マルチキャスト送受信器39、49、59によりマルチキャストネットワーク6から他サーバの構成情報を獲得し、また、マルチキャスト送受信器39、49、59を介して自サーバの構成情報を他のサーバに送出する。

【0022】また、37、47、57はデフォルトのスループット予約テーブルであり、スループット予約テーブル（デフォルト）37、47、57には、システム構成情報、予約情報のデフォルト値が格納されており、システム起動時、スループット予約テーブル37、47、57からデフォルト値が読み込まれ、スループット予約装置38、48、58のスループット予約テーブルに設定される。また、管理者制御インタフェースが設けられており、前記したように予約サーバの管理者は、上記管理者制御インタフェースを使用して、任意のクライアントのスループットを予約することができる。なお、上記デフォルトのスループット予約テーブル37、47、57を書き換えることでスループットを設定することもできる。

【0023】図8は、フロントエンド予約サーバ30、中継サーバ50に設けられるスループット予約テーブルの一例を示す図である。フロントエンド予約サーバ30、中継サーバ50のスループット予約テーブルには、同図に示すように、サーバIDと、上記マルチキャストネットワークを介して受信した、サーバのサービス内容（プロトコル/ホスト名等）、負荷状態（稼働/非稼働状態も含む、例えば非稼働状態のとき"-1"を設定）、および、バックエンド予約サーバ40から送られてくるスループット予約情報（同図の受信）、自サーバで設定されるスループット予約情報（同図の自設定）、および上記情報を受信した時点を示すタイムスタンプ等が格納される。なお、スループット予約情報としては、例えば、サーバ間のスループットのウェイト値、データの転送速度を示すBPS、パケットスイッチの単位時間あたりのスイッチ回数等を設定することができる。

【0024】そして、フロントエンド予約サーバ30、中継サーバ50からバックエンド予約サーバ40にパケ

ットを送出する際には、上記テーブルの予約情報（受信）に基づく転送速度でパケットが送出され、また、フロントエンド予約サーバ30、中継サーバ50からクライアント10側にパケットを送出する際には、上記予約情報（自設定）に基づく転送速度でパケットが送出される。また、バックエンド予約サーバ40のスループット予約テーブルには、リンクされているサーバ名と、自サーバの負荷状態により設定される予約情報（自設定）が格納される。なお、バックエンド予約サーバ40の予約情報は、前記したように、サーバ負荷とサーバスループット特性に基づき動的に設定される。また、前記したように指定された負荷とサーバスループットモデルに基づき設定することも可能であり、上記サーバスループットモデルは前記した管理者制御インタフェースから与えることができる。

【0025】バックエンド予約サーバ40は、上記スループット予約テーブルの予約情報を、パケットスイッチ43、多重化装置44、スループット制御装置B、送受信器Bを経由してネットワークに送り出す。また、この予約情報がフロントエンド予約サーバ30、中継サーバ50で受信されると、受信した予約情報は、送受信器B、スループット制御装置B、多重化装置、パケットスイッチ33、53を経由してスループット予約装置38、58に送られ、スループット予約テーブルに格納される。フロントエンド予約サーバ30、中継サーバ50は、前記したようにこの予約情報（受信）に基づきバックエンド予約サーバ40にパケットを送出する。

【0026】図9～図11は本実施例の処理を示すフローチャートであり、同図を参照しながら、フロントエンド予約サーバ、中継サーバ、バックエンド予約サーバにおける処理について説明する。なお、図9～図11はフロントエンド予約サーバ、中継サーバ、バックエンド予約サーバで共通に使用されるフローチャートとして記述されており、一部の処理はフロントエンド予約サーバ、バックエンド予約サーバ特有の処理である。また、図5、図6、図7中に情報の流れとして矢印で示したa～jは、図9～図11中に示した(a)～(j)に対応している。図9のステップS1において、各サーバにおいて、デフォルトのスループット予約情報、構成情報を設定し、ステップS2において、管理者制御があるかを調べる。管理者制御がある場合には、ステップS3において、管理者制御インタフェースから自サーバのスループット予約、構成情報の設定を行う。また、管理者制御がない場合には、ステップS4に行き、マルチキャスト受信があるかを調べ、マルチキャスト受信がある場合には、ステップS5において、他サーバの構成情報を獲得し、ステップS6において、スループット予約装置内の他のサーバの構成情報テーブルを更新する。

【0027】マルチキャスト受信がない場合には、ステップS7において、マルチキャスト送信があるかを調

べ、マルチキャスト送信がある場合には、ステップS8において、スループット予約装置内の自サーバの構成情報をダンプし、ステップS9において、他サーバへマルチキャスト送信を行う。マルチキャスト受信がない場合には、ステップS10において、送受信器Aで受信があるかを調べ、送受信器Aで受信がある場合（フロントエンド予約サーバの場合にはクライアントからの受信、バックエンド予約サーバの場合にはサーバからの受信）には、ステップS11において、スループット制御装置Aの入力キューに受信したパケットを入れる（図5、図6のa）。送受信器Aで受信がない場合には、図10のステップS12において、送受信器Aで送信があるかを調べ、送受信器Aで送信がある場合には、ステップS13において、スループット制御装置Aの出力キューからパケットを取り出し送信する（フロントエンド予約サーバの場合にはクライアントへの送信、バックエンド予約サーバの場合にはサーバへの送信：図5、図6のb）。

【0028】送受信器Aで送信がない場合には、ステップS14において、送受信器Bで受信があるかを調べ、送受信器Bで受信がある場合（ネットワークからの受信）には、ステップS15において、スループット制御装置Bの入力キューにパケットを入れる（図5、図6、図7のc）。送受信器Bで受信がない場合には、ステップS16において、送受信器Bで送信があるかを調べ、送受信器Bで送信がある場合には、ステップS17において、スループット制御装置Bの出力キューからパケットを取り出し送信する（ネットワークへの送信：図5、図6、図7のd）。送受信器Bで送信がない場合には、図11のステップS18において、スループット制御装置Aからパケットスイッチに入力があるかを調べ、スループット制御装置Aからパケットスイッチに入力がある場合には、ステップS19において、スループット制御装置Aの入力キューからパケットを取り出す（図5、図6のe）。ついで、ステップS20において、スループット予約装置のサーバ構成情報により、パケットの送り先を決め（例えばフロントエンド予約サーバの場合、サービス内容が一致し最も負荷が軽いサーバが選択される）、ステップS21において、多重化装置を介してパケットをスループット制御装置Bの出力キューに入れる（図5、図6、図7のf）。

【0029】スループット制御装置Aからパケットスイッチに入力がない場合には、ステップS22において、スループット制御装置Bから多重化装置を経由してパケットスイッチに入力があるかを調べ、スループット制御装置Bからパケットスイッチに入力がある場合には、ステップS23においてスループット制御装置Bの入力キューからデマルチプレックスしたパケットを取り出す（図5、図6、図7のg）。ついで、ステップS24において、スループット予約装置のサーバ構成情報により、パケットの送り先を決め、ステップS21にお

いて、自サーバ宛のスループット予約であることを調べる。自サーバ宛のスループット予約の場合には、ステップS26において、スループット予約装置で、指定されたスループットを予約する（この処理はフロントエンド予約サーバ、中継サーバのみで行われる）。自サーバ宛のスループット予約でない場合には、ステップS30において、スループット制御装置Aの出力キューにパケットを入れる（フロントエンド予約サーバ、バックエンド予約サーバの場合：図5、図6のh）。あるいは、多重化装置を経由して制御装置Bの出力キューにパケットを入れる（中継サーバの場合：図7のf）。

【0030】スループット制御装置Bから多重化装置を経由してパケットスイッチャに入力がない場合には、ステップS27において、自サーバの負荷変動によりスループット予約が必要であることを調べる（以下の処理はバックエンド予約サーバのみで行われる）。予約が必要である場合には、ステップS28において、スループット予約装置内のスループット予約テーブルを更新する。ついで、ステップS29において、スループット予約装置から、パケットスイッチャ、多重化装置を経由してスループット制御装置Bの出力キューにスループット予約パケットを入れる（図6のj）。以上の処理が終わると、ステップS2に戻り上記処理を繰り返す。なお、上記実施例では、フロントエンド予約サーバ、バックエンド予約サーバを別途設ける場合について説明したが、Internet、ISPなどの既にプロキシサーバを導入しているネットワークでは、既存のプロキシサーバを、上記実施例で示したフロント予約サーバ、バックエンド予約サーバ等で互換性を保ちながら置き換えることもできる。

【0031】

【発明の効果】持続的なWeb通信の急速な増大に対し、サーバ、ネットワークへの投資が追い付かず、恒常的な輻輳状態から脱出できない現状では、何らかの時点でリソース予約システムを導入してユーザ間のリソースの奪い合いを抑制する必要がある。無駄使いをするユーザがより快適になるという、正のフィードバックが掛かり続ければ、イントラネット、インターネットはユーザの強欲によって崩壊してしまう。本発明においては、以上説明したように、クライアントとサーバ間のクライアントの近傍にリソース予約機能を持つフロントエンド・プロキシサーバを設けるとともに、サーバの近傍にリソース割り当てを行うバックエンド・プロキシサーバを設け、負荷状況に応じて動的に各クライアントに対してスループットを予約し、ユーザ単位で平等にリソースを割

り当てるようにしたので、上記正のフィードバックを断ち切ることができ、サーバ負荷の最適化を図ることができる。

【図面の簡単な説明】

【図1】本発明の原理構成図である。

【図2】サーバスループットの特性を示す図である。

【図3】本発明の実施例のシステム構成図である。

【図4】本発明の実施例のシステム構成図（中継サーバを設けた場合）である。

10 【図5】フロントエンド予約サーバのブロック構成図である。

【図6】バックエンド予約サーバのブロック構成図である。

【図7】中継サーバのブロック構成図である。

【図8】スループット予約テーブルの構成例を示す図である。

【図9】スループット予約サーバの動作を説明するフローチャート（1）である。

20 【図10】スループット予約サーバの動作を説明するフローチャート（2）である。

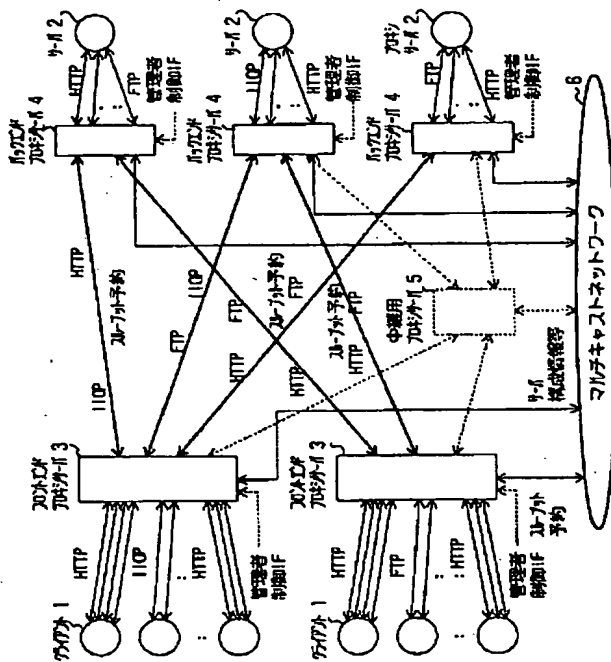
【図11】スループット予約サーバの動作を説明するフローチャート（3）である。

【符号の説明】

1	クライアント
2	サーバ
3	フロントエンド・プロキシサーバ
4	バックエンド・プロキシサーバ
5	中継用プロキシサーバ
6	マルチキャストネットワーク
30 10	Webクライアント
20	Webサーバ
30	フロントエンド予約サーバ
40	バックエンド予約サーバ
50	中継サーバ
31, 41	送受信器A
36, 46, 56	送受信器B
32, 42	スループット制御装置A
35, 45, 55	スループット制御装置B
34, 44, 54	多重化装置
40 33, 43, 53	パケットスイッチャ
38, 48, 58	スループット予約装置
39, 49, 59	マルチキャスト送受信器
37, 47, 57	スループット予約テーブル（デフォルト）

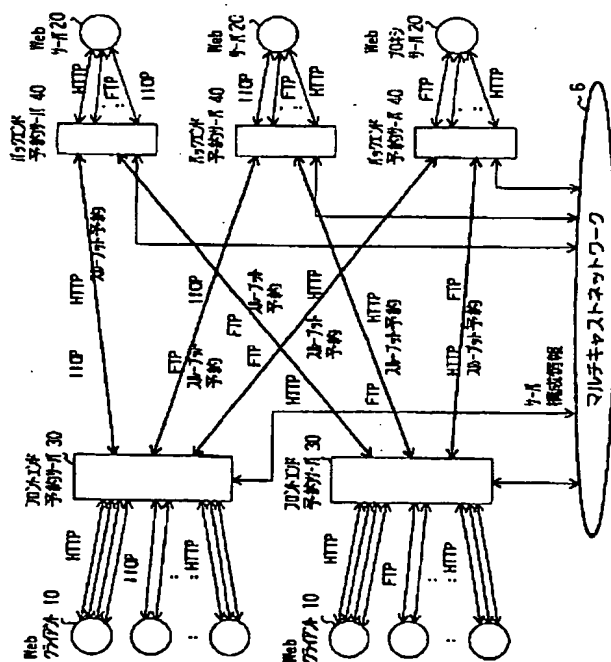
【図1】

本発明の原理構成図



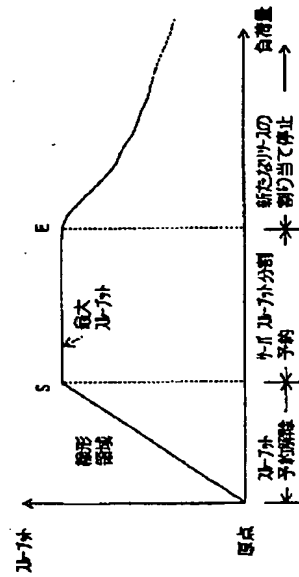
【図3】

本発明の実施例のシステム構成図



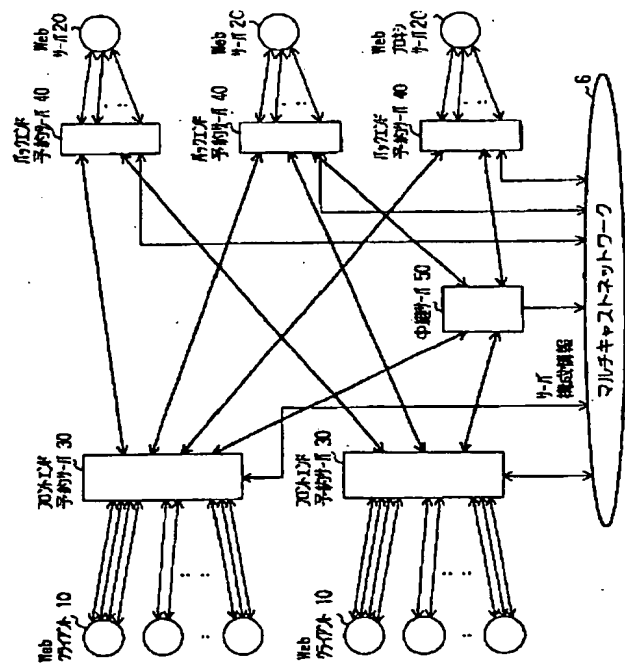
【図2】

サーバスループット特性を示す図



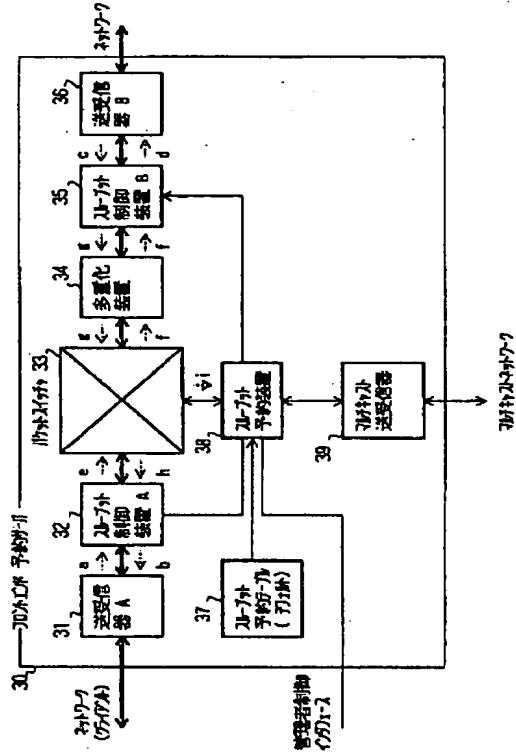
【図4】

本発明の実施例のシステム構成図（中継サーバを設けた場合）



【图 5】

フロントエンド予約サーバのブロック構成図



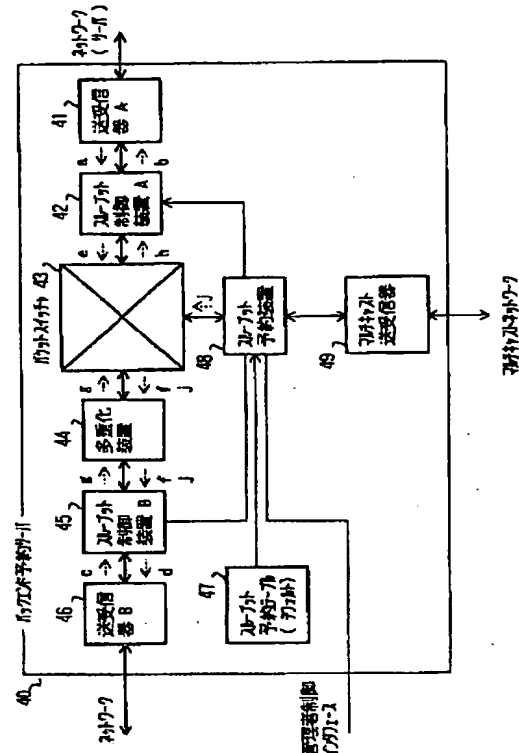
【图 8】

スループット予約テーブルの構成例を示す図

9-11 ID	9-12 内容 (7桁ID/ホ-名)	9-1 自荷 (-1:非接続)	5桁情報 自約定 受信	9-13 力 (日時分秒)
0001	HTTP, ...	L1	P#1	98/03/12, 18:21:10.00
0002	FTP, ...	L2	P#2	:
0003	1/0P, ...	L3	P#3	:
	:	:	:	:
	:	:	:	:
	:	:	:	:

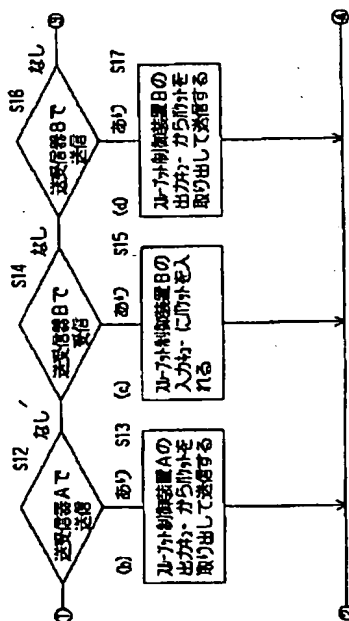
【図 6】

バックエンド予約サーバのブロック構成図



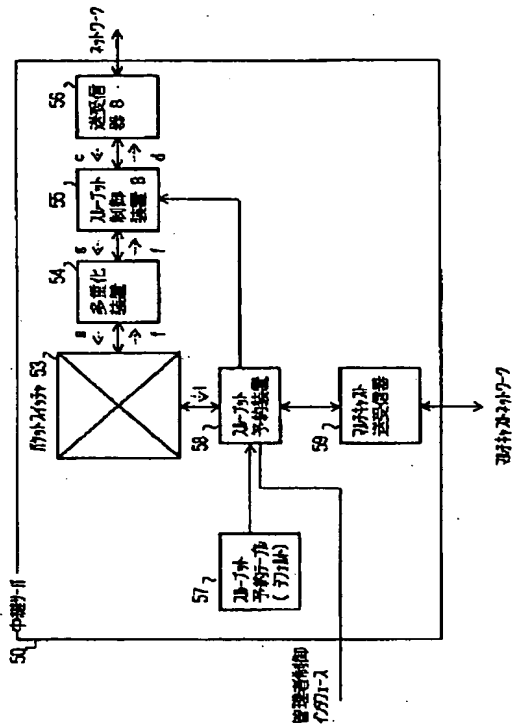
【図 10】

スループット予約サーバの動作を説明するフローチャート(2)



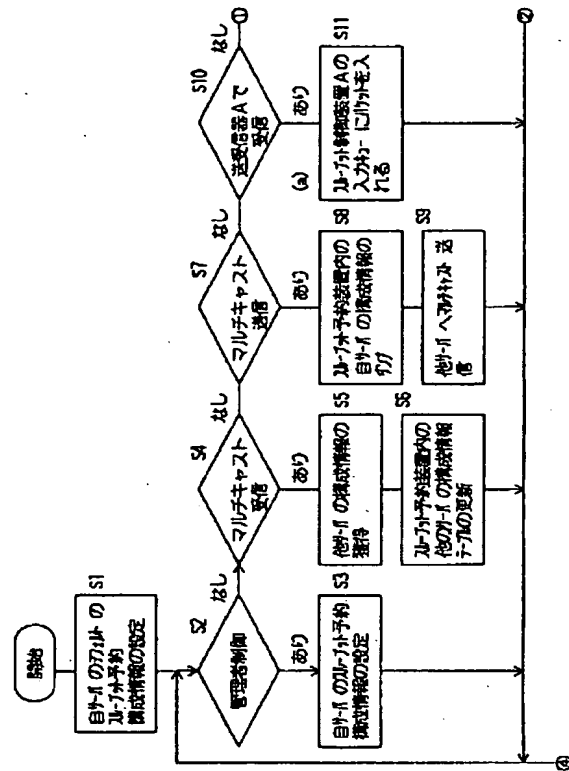
【図 7】

中継サーバのブロック構成図



【図 9】

スループット予約サーバの動作を説明するフローチャート (1)



スループット予約サーバの動作を説明するフローチャート(3)

